ENSC813 Final Project Report
Classical Japanese Character Translator

Cheng-Lin Wu
Student No.301606107
Department of Engineering Science
Simon Fraser University

Abstract

This project focuses on the development of a deep learning-based translator for classical Japanese characters(Kuzushiji) with the aim of revitalizing access to ancient Japanese literature. Despite the significance of Kuzushiji in Japanese cultural heritage, its recognition is limited to a few experts. Leveraging deep learning techniques, our goal is to not only recognize Kuzushiji characters but also transform them into modern Japanese handwriting, facilitating accessibility and comprehension of classical literature. To achieve this, we incorporate the self-attention mechanism into the VGG-16 model for Kuzushiji recognition and employ the diffusion probability model as a generator to create modern handwritten characters. Our results demonstrate that a deeper VGG-16 model with an attention mechanism achieves optimal accuracy in Kuzushiji recognition. Furthermore, our diffusion model successfully generates realistic images of handwritten characters, bridging the gap between classical and modern Japanese script.

1 Background and Motivation

1.1 Obsolete Japanese Character System

In Japanese, Kuzushiji(くずし字) means "deformed characters," which are distorted shapes of Chinese characters. Kuzushiji had been used for over 1,000 years until the Japanese government standardized the character system in the late 19th century. The examples in the following figures are the Kuzushiji in old Japanese literature.

Figure 1: Example of a Kuzushiji literature scroll, Genjimonogatari Uta Awase

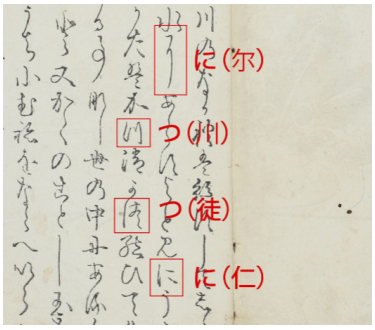

Figure 2: Example of Kuzushiji literature and corresponding modern characters. [1]

Although the modern character system has been introduced for less than 200 years, most native Japanese nowadays cannot recognize Kuzushiji anymore.

1.2 Motivation

Due to the huge volume of classical Japanese literature written in Kuzushiji. A tool to efficiently comprehend Kuzushiji is critical for research and public education. Therefore, the goal of the project is twofold: First, we aim to utilize deep learning techniques to recognize Kuzushiji. Second, to enable a seamless viewing experience for scenarios such as museum exhibitions, we further convert the recognized Kuzushiji into a handwritten modern character.

2 Related Work

Although there is a long history of Kuzushiji studies, its research in the machine learning area has been recently enabled after Clanuwat et al. [2] introduced three comprehensive Kuzushiji datasets, namely Kuzushiji-MNIST, Kuzushiji-49, and Kuzushiji-Kanji. Part of our work is also based on one of the datasets to train the Kuzushiji recognizer.
After the availability of the Kuzushiji datasets, Lyu, Bing, et al. [4] utilize the K-Means algorithm to detect Kuzushiji characters' bounding box from a given page and use the Xception model (a CNN-based neural network) to classify individual identify. Clanuwat et al.[3] propose the Kuronet framework, which is based on UNet architecture, to jointly locate and identify all the Kuzushiji from an old literature page.
Compared to prior work, our work focuses on recognizing a single Kuzushiji character due to the limitation of the collected dataset. But we add value for practical use by further converting the recognized Kuzushiji into modern Japanese handwriting.

3 Problem Definition

This project has two parts. The recognizer module takes the Kuzushiji image as input and predicts the corresponding class. Then, the generator module takes the class as input and generates the corresponding image of a handwritten modern character. To evaluate the performance of Kuzushiji recognition, we measure the accuracy of prediction on the test set.

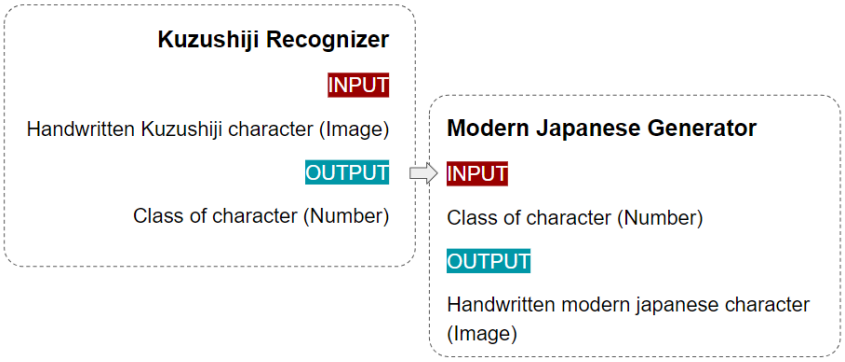

Figure 3: The problem we study has two parts: the recognizer and the generator.

4 Dataset

We use two different Japanese datasets to train the recognizer and generator, respectively. The dataset for Kuzushiji recognition is *Kuzushiji-49* [2]; the name comes from the 49 character classes it contains. There are 232,365 data examples in the training set and roughly 38,547 data examples in the test set. Each example is a 28 by 28 grayscale image, similar to the well-known MNIST dataset.
The dataset for training our generator is *ETL character database* [5]. This dataset is maintained by the Japan Electronics and Information Technology Industries Association; we requested the key to get

access. The dataset has 7,360 data examples, each of them is a 32 x 32 binary image of a handwritten modern Japanese character.
Figure 4 shows some examples from both datasets. On the left are Kuzushiji characters. On the right are modern Japanese characters. We can observe that Kuzushiji's characters are far different from those of the present day. Also, Kuzushiji can use various writing styles for a single class.

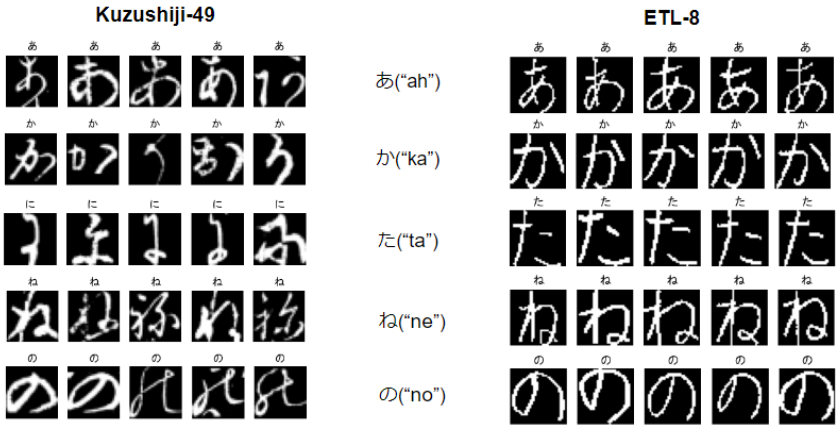

Figure 4: First five data examples for some character classes.

5 Kuzushiji Recognizer

5.1 Solution

First Attempt To train the recognizer, we notice that prior work primarily uses CNN-based models. So, our first attempt is to build a VGG-16 model [6], which can be seen as a go-to CNN-based solution. We train two variants of the VGG-16 model illustrated in Figure 5; one follows the default settings for each convolutional layer, and the other one is a compact version of VGG16, which uses far fewer kernels. The reason is that we think Kuzushiji's character is fairly simple in structure. Also, for a 3 by 3 convolution kernel, the orientation of the character stroke is either in the directions of upper-right, up, upper-left, and so on. Therefore, we assume that kernel numbers in the compact VGG-16 are sufficient to capture all the character features.

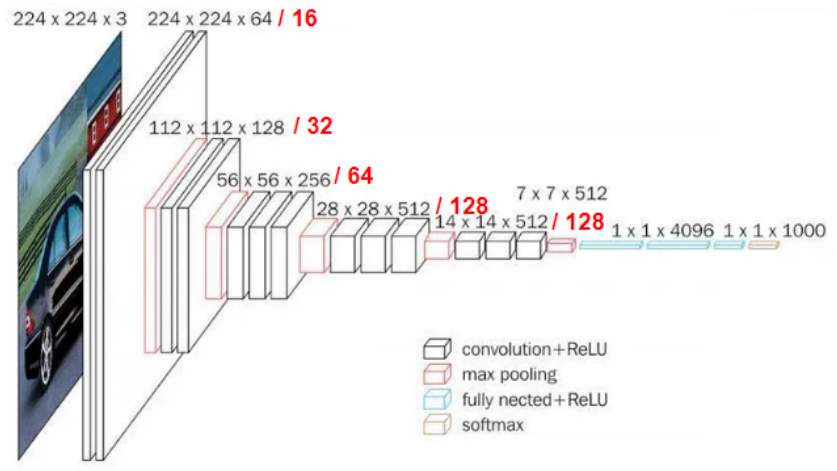

Figure 5: VGG-16 model for Kuzushiji recognition. Numbers in red are the number of kernels for the compact variants.

For training, we use the Adam optimizer [7] on cross-entropy loss.

Second Attempt Through the observation of data examples, we realize that not every region of the image has the same importance. Instead, we should focus more on the central area. To make our model capable of applying different levels of focus on each region, we think of incorporating the self-attention mechanism into our original VGG-16 solution. Therefore, we follow the architecture of AF-CNN proposed by Chen, Qi, et al. [8]. In this model, spatial attention is applied to the last three CNN layers, and the results will be concatenated in the final feedforward layers. The model architecture is illustrated in Figure 6. Likewise, We also fit two variants of its kernel number, which are standard settings of VGG-16 and a compact version.

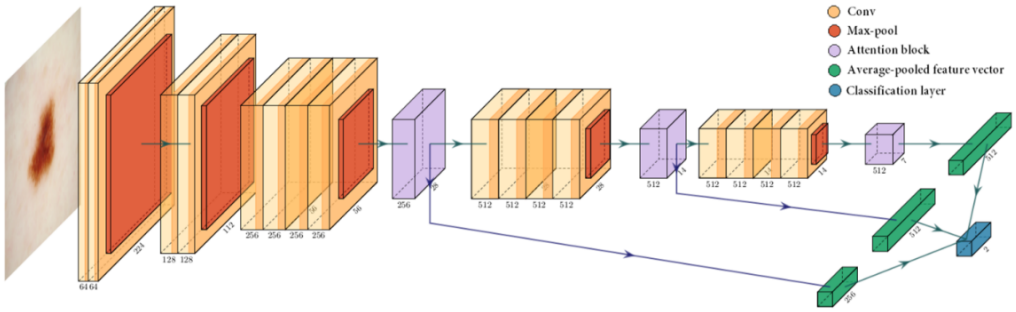

Figure 6: Architecture of AF-CNN[8]

5.2 Results

Table 1 shows the accuracy performance of each recognizer variant. From the result, we observe that the standard VGG-16 with self-attention shows the best accuracy. However, the testing result also indicates that increasing the number of convolution kernels is more effective than adding self-attention because the amount of kernel means the number of features being captured.

Table 1: Comparison of recognition performance

Recognizer settings	Accuracy↑
Compact VGG-16	92.31%
Standard VGG-16	94.63%
Compact VGG-16 with self-attention	92.58%
Standard VGG-16 with self-attention	95.07%

6 Modern Character Generator

6.1 Solution

To generate modern handwritten characters, we hope the output handwritten characters have slightly different styles and strokes. That little nuance can make the handwriting look more realistic. To achieve the goal, we develop a conditional diffusion model using UNet architecture. The diffusion model [9] can generate images from pure random noise, and the term "conditional" means guidance will be provided for the generation; otherwise, the diffusion can generate any character. UNet [10] is an encoder-decoder model that is suitable for image diffusion.

Forward Diffusion Process As depicted in Figure 7, the forward process of the diffusion model is to gradually add noise to the original image (x_0) in T steps until the image becomes full of noise (x_T).
The function $q(x_t|x_{t-1})$ is the forward operation in each timestep. It can be defined as normal distribution:

$$q(x_t|x_{t-1}) = N(x_t, \mu_t = \sqrt{1-\beta_t}x_{t-1}, \sigma_t = \beta_t I)$$

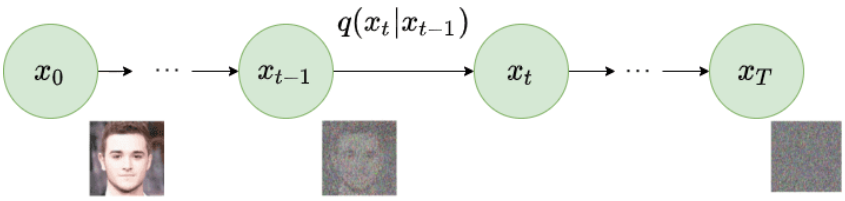

Figure 7: Forward process of diffusion probability model [9].

where β_t is a hyperparameter linearly interpolated from 0.0001 to 0.02 along the change of t. We follow the parameter settings used in the original paper of diffusion model [9]. Over multiple timesteps, the mean of the image will become close to 0, and the whole image will become full of random, uncorrelated noise.

Reverse Diffusion Process The reverse process of the diffusion model is to gradually remove noises from a fully-noised image (x_T) in T steps until the image becomes the original image(x_0). The process is visualized in Figure 8. The function $p_\theta(x_{t-1}|x_t)$ is the reverse function to denoise the image from the previous step. And this function will also be our generator to create a character image from random noise. Therefore, θ is the set of function coefficients (model weights of UNet [10] in our case) we need to train.

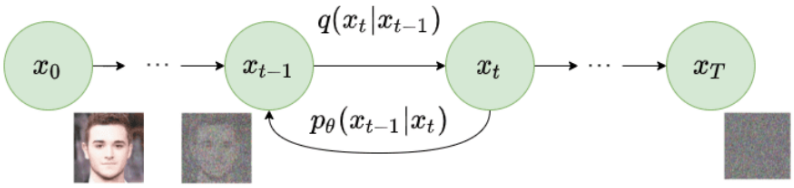

Figure 8: Reverse process of diffusion probability model [9].

The original training objective is to minimize the KL divergence between the original image and the reconstructed image. However, Ho et.al [9] made a few simplifications using re-parameterization tricks and the concept of evidence lower bound (ELBO). The simplified version outperforms the full objective. After the simplification, the training objective becomes to minimize the mean square error between the noise added and the noise predicted.

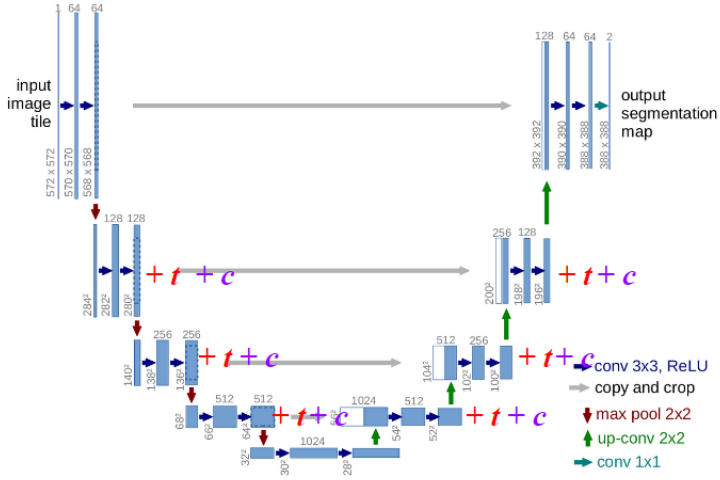

Figure 9: Adding guidance for diffusion model implemented with UNet [10].

Guidance for generation process Without guidance, any characters (legit or non-sense) can be generated by our diffusion model. Therefore, we have to pass a condition to guide the generation of the desired character. In practice, we pass the timestep t into each convolution layer of UNet as one extra feature dimension. Here, t can be seen as a condition of timestep. Similarly, we can pass the class of the desired character c along with timestep t to guide the generation process.

6.2 Results

Figure 10 is the snapshots collected during the model training. The grid on the left is the guidance we pass to the conditional diffusion model, and on the right are the generated handwritings from the first epoch to the last epoch. We can observe that more and more character features are being captured over time.

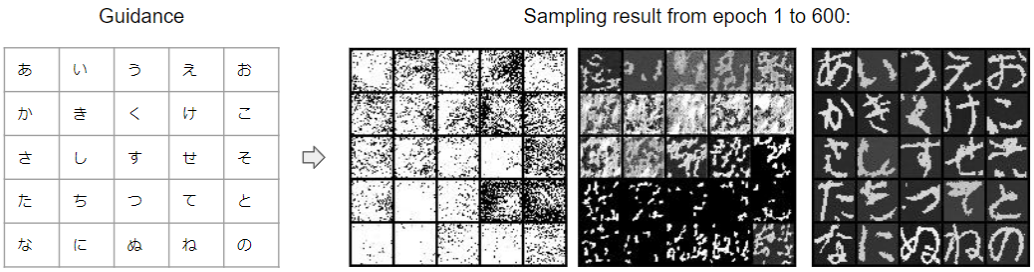

Figure 10: sampling results during generator training.

Finally, Figure 11 is the demo of combining the Kuzushiji recognizer with the modern character generator. Given images of Kuzushiji, the recognizer can identify the class of characters. Then, the generator takes the prediction as input and generates the handwritten modern character from random noise. In Japanese, this sentence is “おはよう”, which means "good morning" in English.

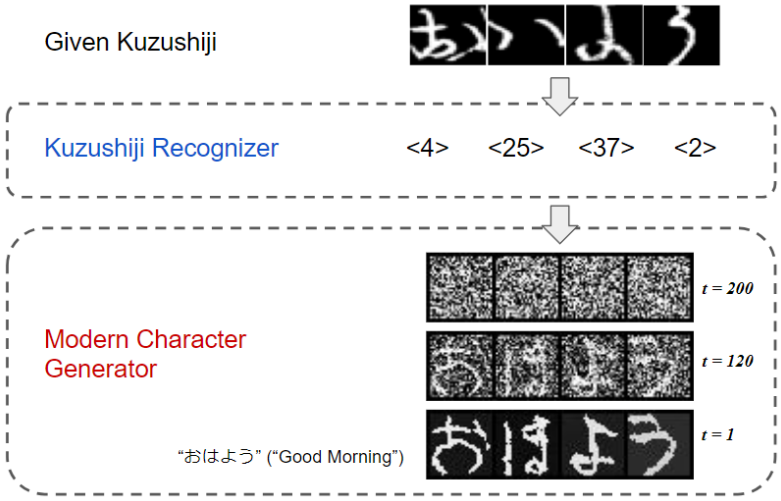

Figure 11: Demo - Kuzushiji recognizer and modern Japanese generator

7 Future Work

During the training of our generator module, we observe that the training loss did not smoothly decrease over the epoch. One possible explanation is that the model is close to convergence at that time, so the impact of outliers (e.g., some special style of character strokes) becomes apparent. To

smooth the sensitivity to outliers, we can employ more regularization techniques or exponential moving average(EMA) on model weight updates.

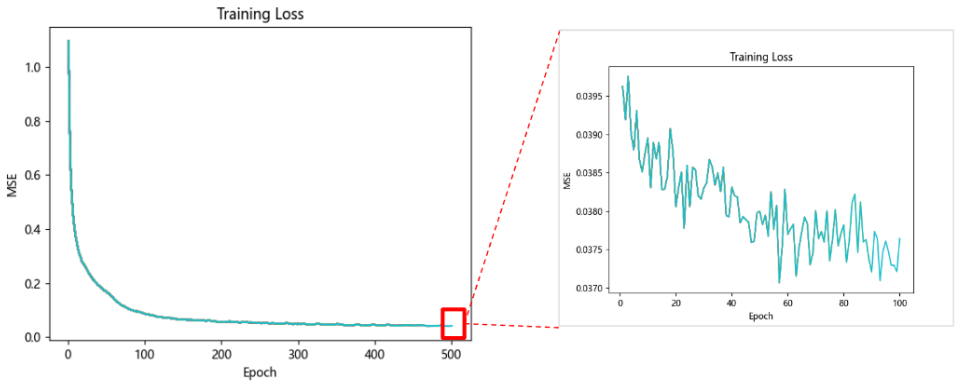

Figure 12: Unstable loss curve during training generator module

8 Conclusion

Our project develops a deep learning-based translator of Kuzushiji characters to make old Japanese literature more accessible. We use deep learning to recognize Kuzushiji characters and turn them into modern Japanese handwriting. This makes it easier for people to read old literature. In practice, We improve the VGG-16 model by adding self-attention and use a diffusion model to create modern characters. Our results show that this improved VGG-16 model works best for recognizing Kuzushiji. Our diffusion model is capable of creating realistic modern characters, helping to bridge the gap between classical and modern Japanese writing. For future improvement, we can incorporate more regularization into diffusion model training to yield a better generation outcome.

References

[1] University of Zürich. “Reading a source in kuzushiji 1 – solution” adfontes.uzh.ch. https://www.adfontes.uzh.ch/en/385103/training/kuzushiji/introduction-kuzushiji/reading-kuzushiji-1-solution (accessed Mar. 30, 2024)

[2] Clanuwat, Tarin, et al. "Deep learning for classical japanese literature." arXiv preprint arXiv:1812.01718 (2018).

[3] Clanuwat, Tarin, et al. "Kuronet: Pre-modern Japanese kuzushiji character recognition with deep learning." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

[4] Lyu, Bing, et al. "The early Japanese books reorganization by combining image processing and deep learning." CAAI Transactions on Intelligence Technology 7.4 (2022): 627-643.

[5] Japan Electronics and Information Technology Industries Association "ETL Character Database". http://etlcdb.db.aist.go.jp/ (accessed Mar. 27, 2024)

[6] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[7] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[8] Chen, Qi, et al. "An Attention-based Convolutional Neural Network for Melanoma Recognition." Journal of Physics: Conference Series. Vol. 1861. No. 1. IOP Publishing, 2021

[9] Ho, Jonathan, Ajay Jain, and Pieter Abbeel. "Denoising diffusion probabilistic models." Advances in neural information processing systems 33 (2020): 6840-6851.

[10] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, 2015.